Attorney's Docket No.: 10559/261001/P9032

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: APPARATUS AND METHOD OF DEVELOPING SOFTWARE FOR A MULTI-PROCESSOR CHIP

APPLICANT: **WILLIAM R. WHEELER, LAI-WAH HUI, DONALD F. HOOPER, SERGE KORNFELD AND JAMES D. GUILFORD**

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL584779493US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

December 21, 2000
Date of Deposit

Signature

Derek Norwood
Typed or Printed Name of Person Signing Certificate

APPARATUS AND METHOD OF DEVELOPING SOFTWARE FOR A MULTI-PROCESSOR CHIP

BACKGROUND OF THE INVENTION

Field of The Invention:

The present invention relates generally to the field of digital data processing systems and, more particularly, to an apparatus and method of developing applications software for a multi-processor chip.

Background:

The apparatus and method of developing software for a multi-processor chip provides a solution to developing applications software for a new multi-processor chip design, with fast simulation.

Conventional development of applications software for a new processor chip requires a software simulation model of the entire new chip design, and then the applications software is run on the model of the chip. Simulation in this way may be slow, due to the sheer quantity of design states that must be modeled for the chip.

As integrated circuit geometries scale with process technology, more area becomes available on a chip for new functions. New multi-processor chip designs are beginning to integrate processor designs from previous production processor chips, where a previous production processor is essentially lifted and inserted into a corner of the new multi-processor chip. The previous processors have highly detailed simulation models, such as a detailed gate-level model. Because the simulation models are highly detailed, the processor simulation models are highly computationally intensive and therefore relatively slow to run.

Newer processor designs are known that have two simulation models. One version of the simulation model is the previously described detailed simulation model. The other version is a high-level, fast, cycle-accurate model. By simulating with the high-level model, high-speed simulation of the chip designs can be achieved.

When new multi-processor chips are designed that use previously designed processors and newer processors on one chip, high-speed simulation of the entire multi-processor chip may not be achieved because no high-speed simulation model exists for the previous processors. This causes a bottleneck in the development of the applications software.

Further, because the new multi-processor chips contain multiple processors, communications need to be tested between the processors, which involve interrupt and signaling mechanisms. The previous design chip sections do not have this hardware.

Another problem is simulation of memory access. Initialization of tables in memory can take as long as forty-five minutes. A need exists to circumvent this startup delay for most tests.

A need therefore exists for a solution to develop and test applications software on a new multi-processor chip design that uses previous processors and newer processors, with fast simulation and minimal resources expended to create the simulation model. Furthermore, the solution should provide a way to test communications between processors and fast simulation of memory access.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a full hardware configuration of a multi-processor chip configured according to the present invention.

FIG. 2 is a block diagram of a full software configuration of the multi-processor chip configured according to the present invention.

FIG. 3 is a block diagram of a partial hardware configuration the multi-processor chip configured according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention, which tends to address this need, resides in an apparatus and method of developing applications software for a multi-processor chip. The apparatus and method described herein provides advantages over known solutions to develop and test applications software for a multi-processor chip design in that it provides fast simulation and minimal resources expended to create the simulation model of the chip design. Furthermore, the apparatus and method provides a way to test communications between processors, fast simulation of memory access, and easy configuration to test selected applications software.

According to the present invention, applications software can be rapidly tested and developed for a multi-processor chip. This can be accomplished by executing

software simulations of the new processors and applications software on a previously designed processor that is part of the hardware on the multi-processor chip. The previously designed processor does not need to be simulated as part of the multi-chip simulation, which may speed up the simulation. The previously designed processor is typically much faster than those found on a personal workstation development platform, and therefore the execution time is faster. The processor simulation and application software can be configured to take advantage of the configuration most appropriate for fast execution, and avoid simulation of portions of the new processors not necessary for testing the applications software. The second processor corresponding to the applications software being debugged can be set in slow simulation mode while the other second processors can be set in fast simulation mode.

Thus, the software can be developed with fast simulation and minimal resources. Furthermore, the apparatus and method provides a way to test communications between processors, fast simulation of memory access, and easy configuration to test selected applications software.

FIG. 1 illustrates a full hardware configuration of a multi-processor chip 10 configured according to the present invention. Ultimately, a multi-processor chip under

development will have processors 1 through N in hardware on a single silicon chip. According to the present invention, one of these processors 12 can be a previously designed, production processor, such as, the StrongARM™ brand processor available from Intel, Inc. The other processors 14 can be newly designed processors that are part of the hardware on the single silicon chip. In an embodiment, the multi-processor chip is designed for internetworking applications, but a skilled artisan will recognize that other applications may employed according to the invention.

The first processor 12 has an embedded operating system 16 that will run on the production chip, such as, VXWORKS, UCOS, PSOS, OS9, or LINUX. Furthermore, first processor 12 includes an operating system interface 18, which interfaces with a first applications software 20.

Applications software 1 through N can be applications software that execute on processors 1 through N, respectively. The first applications software 20 executes on the first processor 12 and has procedures that communicate with other portions of the multi-chip processor. Furthermore, the first applications software 20 interfaces to the rest of the chip through a hardware abstraction layer (HAL) 22, which is a set of function calls.

Applications software 2 through N (24) run on processors 2 through N, respectively. In an embodiment, applications software 2 through N can be programs to perform processing of packets or frames in Frame Relay and Ethernet applications.

According to the present invention, a variety of configurations for the operating system, the processor simulation, and the applications software are available to facilitate the development of the applications software for the multi-processor chip.

For example, FIG. 2 is a block diagram of a full software configuration of the multi-processor chip 10 configured according to the present invention. Software simulations 26 of one or more second processors 14 that are to be provided in hardware on the single silicon chip are loaded on a personal workstation development platform 28. One or more applications software 24 that are to be executed on the hardware 14 of the corresponding second processors also are loaded on the platform. The software simulations of the second processors and the corresponding applications software are executed on the workstation platform using the processor and operating system 30 of the workstation.

In the full software simulation configuration, the simulation runs on a different operating system than the embedded operating system 16 of the first processor 12. For example, the non-embedded operating system can be Microsoft Windows NT. Windows NT may be preferred because of the convenience of the NT development platform.

The operating system interface 18 provides a simple generic interface to first applications software 20, while translating the system calls to those of the non-embedded operating system 30. Furthermore, the operating system interface simulates interrupts from the second processor simulations 26 to the first processor, thus providing a way to test communications between the processors.

The applications software 24 includes an interface that allows software to be configured to execute on either the hardware of the second processor 14 or to execute with the simulation of the second processor 26. In addition, each software simulation includes a slow, highly detailed software simulation of the second processor and a fast, high-level simulation of the second processor. The interface additionally allows the software simulation to be selected as either the slow, highly detailed software simulation or the fast, high-level simulation.

In the slow, highly detailed software simulation, the first applications software 20 communicates through a simulation of the intra-chip bus interface, which is part of the processor simulation 26, to the second processor simulation 26 and to memory and control status registers. In the fast, high-level simulation, the bus interface is bypassed. This latter mode may be used to perform fast accesses and large array accesses in the simulator.

This configuration can also be advantageously used to test the portable language portions of first applications software 20.

Another example of the variety of possible configurations to develop the applications software is illustrated in FIG. 3. FIG. 3 is a block diagram of a partial hardware configuration of the multi-processor chip configured according to the present invention. In the partial hardware configuration, the applications software 20, 24 and software simulations 26 are executed on first processor 12, which is provided as hardware on the single silicon chip 10. The software simulations of one or more second processors that are to be provided in hardware on the single silicon chip are loaded on the first processor. One or more applications software 24 that are to be

executed on the hardware of the corresponding second processors also are loaded on the first processor.

The applications software 24 includes an interface that allows the software to be configured to execute on either the hardware of the second processor or to execute with the simulation of the second processor on the first processor 12. In addition, the software simulation of the second processor includes a slow, highly detailed software simulation of the second processor and a fast, high-level simulation of the second processor. The interface additionally allows the software simulation to be selected as either the slow, highly detailed software simulation or the fast, high-level simulation.

This configuration provides the advantage of being able to test and develop the applications software using the actual hardware of the first processor and the embedded operating system, and thus avoid the slow simulation of the first processor and reduce the number of ancillary bugs. Further, this configuration uses the high-processing speed of the first processor as opposed to a slower workstation platform.

Further, only the second processor corresponding to the applications software being debugged needs to be set in slow simulation mode while the other second processors can

be set in fast simulation mode. This too results in faster execution times of the simulation.

As hardware becomes available for the second processors, the interface of the corresponding applications software can be configured to run on the hardware, thus improving the speed of processing by avoiding the slower software simulation of the second processor.

This configuration can also be advantageously used to test the non-portable language portions of the first applications software.

In conclusion, the apparatus and method of developing software described herein provides advantages over known solutions to develop and test applications software for a multi-processor chip design in that it provides fast simulation and minimal resources expended to create the simulation model of the chip design. Furthermore, the apparatus and method provides a way to test communications between processors, fast simulation of memory access, and easy configuration to test selected applications software. This is primarily accomplished by utilizing the speed of a previously designed high-speed processor to run the simulations of new processor designs and applications software and utilizing the high-speed and low-speed simulation of the new processor designs where appropriate.

Those skilled in the art will recognize that other modifications and variations can be made the method of developing software of the present invention without departing from the scope or spirit of this invention.